UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/821,774 | 04/08/2004 | Calum Murray | 16319-08293 | 1447 |

758 7590 05/21/2007

FENWICK & WEST LLP
SILICON VALLEY CENTER
801 CALIFORNIA STREET
MOUNTAIN VIEW, CA 94041

| EXAMINER | |
|---|---|
| LOUIE, OSCAR A | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2109 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 05/21/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/821,774 | MURRAY ET AL. | |
| | Examiner | Art Unit | |
| | Oscar A. Louie | 2109 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *08 April 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 April 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *10/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

This first non-final action is in response to the original filing of 04/08/2004. Claims 1-39 are pending and have been considered as follows.

### *Specification*

1. The disclosure is objected to because of the following informalities:

- Paragraph 0004 line 8 page 2 contains the term "comprising," which is an obvious typo and should be replaced with the term "compromising" to fit the language of the sentence.
- Paragraph 0047 line 8 page 17 contains the acronyms "EPROM" and "EEPROM" but have not properly defined them.

Appropriate correction is required.

### *Claim Objections*

2. Claims 4 & 10 are objected to because of the following informalities: Line 1 of both claims disclose the portion, "the decrypting the database," however, the grammar should read as such "the decryption of the database." Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 35-39 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

- Claim 35 recites the limitation "the encryption data file" in line 3. Claim 35 is a dependent claim off from Claim 34 which recites no such limitation. There is insufficient antecedent basis for this limitation in the claim.
- Claim 37 recites the limitation "the encrypted password" in line 6. Claim 37 is an independent claim which possesses no such prior limitation. There is insufficient antecedent basis for this limitation in the claim.
- Claim 38 recites the limitation "the computer program product" in line 1. Claim 38 is a dependent claim off from Claim 37 which recites no such limitation. There is insufficient antecedent basis for this limitation in the claim. It is noted by the examiner that this particular portion of Claim 38 appears to be a typographical error and should be read as "the method of claim 37" instead of, "the computer program product of claim 37."

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lampson et al ("Authentication in distributed systems: theory and practice").

Claim 1:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, but do not explicitly disclose,

- "reading a file dumped from a database, the file containing an encrypted database password"
- "decrypting the database password"
- "initiating a user session with the database"

however, Lampson et al do disclose,

- "If we have requests $K_{abadt}$ says "read from foo" and $K_{burrows}$ says "read from foo", and file foo has the ACL SRC $\Lambda$ Manager, we must get from $K_{abadt}$ => Abadt => SRC and $K_{burrows}$ => Burrows => Manager to $K_{abadt}$ $\Lambda$ $K_{burrows}$ => SRC $\Lambda$ Manager. This lets us reason from the two requests to SRC $\Lambda$ Manager says "read from foo", and the ACL obviously grants this" [page 168 column 1];

- "The receiver needs to know what key Kit should use to decrypt a message. If K is a public key we can send it along with the encrypted message; all the receiver has to do is check that K actually decrypts the message correctly" [page 170 column 1];
- "A similar scheme handles delegation from the user U to the workstation IV on which she logs in" [page 177 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "reading a file dumped from a database, the file containing an encrypted database password" and "decrypting the database password" and "initiating a user session with the database," in the invention as disclosed by Lampson et al since it is common to read information from a file or some other form of organized data. It is also common to have to encrypt and decrypt information, particularly sensitive information (i.e. usernames and passwords) that may grant access to data centers (i.e. databases). User sessions are also common between a user and a workstation or any other network entity (i.e. database).

Claims 7 & 25:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor and a method for controlling a processor to connect to a database, but do not explicitly disclose,

- "initiating a signon attempt to the database, the signon attempt programmably failing to connect, the signon attempt triggering an embedded mechanism that dumps an encrypted database password into a file"

- "reading the file"
- "decrypting the database password"
- "initiating a user session with the database"

however, Lampson et al do disclose,

- "We therefore use the joint authority rule (Pl 2) to make this delegation require a countersignature by a temporary public key K1. This key is made at login time and called the login session key...If there is a threat that the workstation might be compromised within 30 minutes after a logout, then it should also discard its master key and node key at logout" [page 177 column 1];
- "If we have requests $K_{abadt}$ says "read from foo" and $K_{burrows}$ says "read from foo", and file foo has the ACL SRC $\Lambda$ Manager, we must get from $K_{abadt}$ => Abadt => SRC and $K_{burrows}$ => Burrows => Manager to $K_{abadt}$ $\Lambda$ $K_{burrows}$ => SRC $\Lambda$ Manager. This lets us reason from the two requests to SRC $\Lambda$ Manager says "read from foo", and the ACL obviously grants this" [page 168 column 1];
- "The receiver needs to know what key Kit should use to decrypt a message. If K is a public key we can send it along with the encrypted message; all the receiver has to do is check that K actually decrypts the message correctly" [page 170 column 1];
- "A similar scheme handles delegation from the user U to the workstation IV on which she logs in" [page 177 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "initiating a signon attempt to the database, the signon attempt programmably failing to connect, the signon attempt triggering an embedded mechanism that dumps an encrypted database password into a file" and "reading the file" and "decrypting the database password" and "initiating a user session with the database," in the invention as disclosed by Lampson et al since a failed log on can be used to "protect the user in case the workstation is compromised after she logs out" [page 177 column 1]. It is common to read information from a file or some other form of organized data, as well as, to encrypt and decrypt information, particularly sensitive information (i.e. usernames and passwords), that may grant access to information data centers (i.e. databases). User sessions are also common between a user and a workstation or any other network entity (i.e. database).

Claims 2, 8, & 26:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor and a method for controlling a processor to connect to a database, as in Claims 1, 7, & 25 above respectively, further comprising,

- "creating a temporary signon during the user session" (i.e. "This key is made at login time and called the login session key. When the user logs out, the workstation forgets K1-1 so that it can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired") [page 177 column 1];

- "initiating a temporary user session with restricted access using the temporary signon" (i.e. "This key is made at login time and called the login session key. When the user logs out, the workstation forgets K1-l so that it can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired") [page 177 column 1].

Claims 3, 9, & 27:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claims 1, 7, & 25 above respectively, further comprising,

- "the database password is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA") [page 169 column 2].

Claims 4, 10, & 28:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claims 1, 7, & 25 above respectively, further comprising,

- "decrypting the database password is accomplished using a private key" (i.e. "Usually K is made public and K-l kept private, so that the holder of K-1 can broadcast messages with integrity") [page 169 column 2].

Claims 5, 11, & 29:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claims 1, 7, & 25 above respectively, further comprising,

- "the database password comprises a hash of a user name and password" (i.e. "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest") [page 169 column 1].

Claims 6, 12, & 30:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claims 1, 7, 25 above respectively, further comprising,

- "passing a connect string to a database tool, the connect string including the database password" (i.e. "This seems reasonable, since getting access to the user's key will require her to type her password or insert her smart card and type a PIN") [page 177 column 1].

Claim 13:

Lampson et al disclose a computer program product for controlling a processor to connect to a database but does not explicitly disclose,

- "a computer readable medium"
- "an attempted signon module stored on the medium, the attempted signon module communicatively coupled to a database to initiate a signon attempt to the database"

- "a read module stored on the medium to read a file dumped by the database, the file containing an encrypted database password"
- "a decryption module stored on the medium to decrypt the database password"
- "a temporary signon module stored on the medium, the temporary signon module communicatively coupled to the database to initiate a user session with the database"

however, Lampson et al do disclose,

- "a cache" [page 167 column 1];
- "We therefore use the joint authority rule (Pl 2) to make this delegation require a countersignature by a temporary public key K1. This key is made at login time and called the login session key...If there is a threat that the workstation might be compromised within 30 minutes after a logout, then it should also discard its master key and node key at logout" [page 177 column 1];
- "If we have requests $K_{abadt}$ says "read from foo" and $K_{burrows}$ says "read from foo", and file foo has the ACL SRC $\Lambda$ Manager, we must get from $K_{abadt}$ => Abadt => SRC and $K_{burrows}$ => Burrows => Manager to $K_{abadt}$ $\Lambda$ $K_{burrows}$ => SRC $\Lambda$ Manager. This lets us reason from the two requests to SRC $\Lambda$ Manager says "read from foo", and the ACL obviously grants this" [page 168 column 1];
- "The receiver needs to know what key Kit should use to decrypt a message. If K is a public key we can send it along with the encrypted message; all the receiver has to do is check that K actually decrypts the message correctly" [page 170 column 1];
- "initiating a temporary user session with restricted access using the temporary signon" [page 177 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "a computer readable medium" and "an attempted signon module stored on the medium, the attempted signon module communicatively coupled to a database to initiate a signon attempt to the database" and "a read module stored on the medium to read a file dumped by the database, the file containing an encrypted database password" and "a decryption module stored on the medium to decrypt the database password" and "a temporary signon module stored on the medium, the temporary signon module communicatively coupled to the database to initiate a user session with the database, in the invention as disclosed by" Lampson et al since a cache storage would "make frequent operations fast" [page 167 column 1]. An attempted sign on module is desirable since a failed log on can be used to "protect the user in case the workstation is compromised after she logs out" [page 177 column 1]. It is common to read information from a file or some other form of organized data, as well as, to encrypt and decrypt information, particularly sensitive information (i.e. usernames and passwords), that may grant access to information data centers (i.e. databases). User sessions are also common between a user and a workstation or any other network entity (i.e. database); particularly temporary sessions which "can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired" [page 177 column 1].

Claim 14:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 13 above, further comprising,

- "the temporary signon module creates a temporary signon during the user session and initiates a temporary user session with restricted access using the temporary signon" (i.e. "This key is made at login time and called the login session key. When the user logs out, the workstation forgets K1-l so that it can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired") [page 177 column 1].

Claim 15:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 13 above, further comprising,

- "the database password is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA") [page 169 column 2].

Claim 16:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 13 above, further comprising,

- "the decryption module stored on the medium to decrypt the database password uses a private key" (i.e. "Usually K is made public and K-l kept private, so that the holder of K-1 can broadcast messages with integrity") [page 169 column 2].

Claim 17:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 13 above, further comprising,

- "the database password is encrypted with a public key" (i.e. "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest") [page 169 column 1].

Claim 18:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 13 above, further comprising,

- "the database password is encrypted with a public key" (i.e. "This seems reasonable, since getting access to the user's key will require her to type her password or insert her smart card and type a PIN") [page 177 column 1].

Claim 19:

Lampson et al disclose a method for controlling a processor to connect to a database, but does not explicitly disclose,

- "executing a launcher program"
- "reading with the launcher program a file dumped from a database, the file containing an encrypted database password"
- "decrypting the database password"
- "initiating a user session with the database"

however, Lampson et al do disclose,

- "With these ideas we can explain exactly how to load a program securely" [page 175 column 2];
- "If we have requests $K_{abadt}$ says "read from foo" and $K_{burrows}$ says "read from foo", and file foo has the ACL SRC $\Lambda$ Manager, we must get from $K_{abadt}$ => Abadt => SRC and $K_{burrows}$ => Burrows => Manager to $K_{abadt}$ $\Lambda$ $K_{burrows}$ => SRC $\Lambda$ Manager. This lets us reason from the two requests to SRC $\Lambda$ Manager says "read from foo", and the ACL obviously grants this" [page 168 column 1];
- "The receiver needs to know what key Kit should use to decrypt a message. If K is a public key we can send it along with the encrypted message; all the receiver has to do is check that K actually decrypts the message correctly" [page 170 column 1];
- "A similar scheme handles delegation from the user U to the workstation IV on which she logs in" [page 177 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "executing a launcher program" and "reading with the launcher program a file dumped from a database, the file containing an encrypted database password" and "decrypting the database password" and "initiating a user session with the database," in the invention as disclosed by Lampson et al since loading a program securely can ensure the execution of another program by managing proper resource allocation. It is common to read information from a file or some other form of organized data, as well as, to encrypt and decrypt information, particularly sensitive information, that may grant access to information data centers (i.e. databases). User sessions are also common between a user and a workstation or any other network entity (i.e. database).

Claim 20:

Lampson et al disclose a method for controlling a processor to connect to a database, as in Claim 19 above, further comprising,

- "creating a temporary signon during the user session" (i.e. "This key is made at login time and called the login session key. When the user logs out, the workstation forgets K1-l so that it can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired") [page 177 column 1];
- "initiating a temporary user session with restricted access using the temporary signon" (i.e. "This key is made at login time and called the login session key. When the user logs out, the workstation forgets K1-l so that it can no longer refresh any credentials that depend on the login delegation, and hence can no longer act for the user after the 30 minute lifetime has expired") [page 177 column 1].

Claim 21:

Lampson et al disclose a method for controlling a processor to connect to a database, as in Claim 19 above, further comprising,

- "the database password is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA") [page 169 column 2].

Claim 22:

Lampson et al disclose a method for controlling a processor to connect to a database, as in Claim 19 above, further comprising,

- "the decrypting the database password is accomplished using a private key" (i.e. "Usually K is made public and K-1 kept private, so that the holder of K-1 can broadcast messages with integrity") [page 169 column 2].

Claim 23:

Lampson et al disclose a method for controlling a processor to connect to a database, as in Claim 19 above, further comprising,

- "the database password comprises a hash of a user name and password" (i.e. "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest") [page 169 column 1].

Claim 24:

Lampson et al disclose a method for controlling a processor to connect to a database, as in Claim 19 above, further comprising,

- "passing a connect string to a database tool, the connect string including the database password" (i.e. "This seems reasonable, since getting access to the user's key will require her to type her password or insert her smart card and type a PIN") [page 177 column 1].

Claim 31:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, but do not explicitly disclose,

- "hashing a client user name and password to create a database password"
- "encrypting the database password to create an encrypted password"
- "storing the encrypted password"
- "receiving a signon attempt at the database"
- "failing the signon attempt"
- "dumping a file containing the encrypted password"

however, Lampson et al do disclose,

- "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest" [page 169 column 1];
- "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA [20]. In this scheme $(K^{-1})^{-1} = K$, so anyone can send a secret message to the holder of $K^{-1}$ by encrypting it with K" [page 169 column 2];
- "The receiver gets them from the sender and caches them" [page 178 column 1];
- "a component of the receiver's operating system called the authentication agent does this work for the receiver" [page 178 column 2];

- "It's not quite true that components outside the TCB can fail without affecting security. Rather, the system is "fail-secure"" [page 167 column 1];
- "the credentials area collection of certificates and statements from the sender, together with the connective tissue that assembles them into a proof of $C_{aid}$ => A. The receiver gets them from the sender" [page 178 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "hashing a client user name and password to create a database password" and "encrypting the database password to create an encrypted password" and "storing the encrypted password" and "receiving a signon attempt at the database" and "failing the signon attempt" and "dumping a file containing the encrypted password," in the invention as disclosed by Lampson et al since hashing (i.e. one way function) of a username and password is common practice in public key encryption, as is caching the encrypted password or information. Logins or user sessions are common between networked entities. Fail-secure systems (i.e. failing a signon attempt) are also common for improved security by not making it obvious to a potential intruder that a particular user account is special in comparison to regular accounts. A typical reaction to a fail-secure system is the sending of encrypted information and/or other credentials (i.e. dumping a file containing the encrypted password).

Claim 32:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claim 31 above, further comprising,

- "allowing access to the database using the database password" (i.e. "The TCB for granting access is just CA; that for revocation is CA and O") [page 172 column 2].

Claim 33:

Lampson et al disclose a computer program product, comprising a computer readable medium storing computer executable instructions for controlling a processor, as in Claim 31 above, further comprising,

- "the encrypted password is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA [20]. In this scheme $(K^{-1})^{-1} = K$, so anyone can send a secret message to the holder of $K^{-1}$ by encrypting it with K") [page 169 column 2].

Claim 34:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, but do not explicitly disclose,

- "a computer readable medium"
- "a hash module stored on the medium to hash a user name and password to create a database password"

- "an encryption module stored on the medium to encrypt the database password to create an encrypted database password"
- "a store module stored on the medium, the store module communicatively coupled to a database to store the encrypted database password in a file accessible independently of the database"

however, Lampson et al do disclose,

- "a cache" [page 167 column 1];
- "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest" [page 169 column 1];
- "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA [20]. In this scheme $(K^{-1})^{-1} = K$, so anyone can send a secret message to the holder of $K^{-1}$ by encrypting it with K" [page 169 column 2];
- "The receiver gets them from the sender and caches them" [page 178 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "a computer readable medium" and "a hash module stored on the medium to hash a user name and password to create a database password" and "an encryption module stored on the medium to encrypt the database password to create an encrypted database password" and "a store module stored on the medium, the store module communicatively coupled to a database to store the encrypted database password in a file accessible independently of the database," in the invention as

disclosed by Lampson et al since a cache is a form of computer readable medium that may be used to store encrypted information. Encrypted information is typically hashed together when it is a combination of pieces of information (i.e. username and password) prior to performing the actual encryption process.

Claim 35:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 34 above, further comprising,

- "a send module stored on the medium, the send module communicatively coupled to a launcher application to send the encrypted data file to a launcher application" (i.e. "Our system implements remote procedure call, so it has call and return messages. For a call, statements are made by the caller (the client) and interpreted by the called procedure (the server); for a return, the reverse is true.") [page 177 column 2].

Claim 36:

Lampson et al disclose a computer program product for controlling a processor to connect to a database, as in Claim 35 above, further comprising,

- "the encrypted data file is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA") [page 169 column 2].

Claim 37:

Lampson et al disclose a method for controlling a processor to connect to a database and a launcher application, but do not explicitly disclose,

- "executing a software application"
- "hashing a user name and password to create a database password"

- "encrypting the database password to create an encrypted database password"
- "storing the encrypted password"
- "receiving a signon attempt at the database"
- "failing the signon attempt"
- "dumping a file containing the encrypted password"

however, Lampson et al do disclose,

- "With these ideas we can explain exactly how to load a program securely" [page 175 column 2];
- "For integrity it is enough to encrypt a digest of the message. A digest is the result of a one-way function; this means that you can't invert the function and compute a message with a given digest" [page 169 column 1];
- "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA" [page 169 column 2];
- "The receiver gets them from the sender and caches them" [page 178 column 1];
- "a component of the receiver's operating system called the authentication agent does this work for the receiver" [page 178 column 2];
- "It's not quite true that components outside the TCB can fail without affecting security. Rather, the system is "fail-secure"" [page 167 column 1];
- "the credentials area collection of certificates and statements from the sender, together with the connective tissue that assembles them into a proof of $C_{aid}$ => A. The receiver gets them from the sender" [page 178 column 1];

Therefore, it would have been obvious to one of ordinary skill in the art to include, "executing a software application" and "hashing a user name and password to create a database password" and "encrypting the database password to create an encrypted database password" and "storing the encrypted password" and "receiving a signon attempt at the database" and "failing the signon attempt" and "dumping a file containing the encrypted password," in the invention as disclosed by Lampson et al since loading a program securely can ensure the execution of another program by managing proper resource allocation. The hashing (i.e. one way function) of a username and password is common practice in public key encryption, as is caching the encrypted password or information, prior to performing the process of encrypting the information. Logins or user sessions are common between networked entities. Fail-secure systems (i.e. failing a signon attempt) are also common for improved security by not making it obvious to a potential intruder that a particular user account is special in comparison to regular accounts. A typical reaction to a fail-secure system is the sending of encrypted information and/or other credentials (i.e. dumping a file containing the encrypted password).

Claim 38:

Lampson et al disclose a method for controlling a processor to connect to a database and a launcher application, as in Claim 37 above, further comprising,

- "allowing access to the database using the database password" (i.e. "The TCB for granting access is just CA; that for revocation is CA and O") [page 172 column 2].

Claim 39:

Lampson et al disclose a method for controlling a processor to connect to a database and a launcher application, as in Claim 37 above, further comprising,

- "the encrypted password is encrypted with a public key" (i.e. "The most popular public key encryption scheme is Rivest-Shamir-Adleman or RSA") [page 169 column 2].

***Conclusion***

7. The prior art made of record and not relied upon is considered pertinent to the applicant's disclosure.

a. Ando et al (US-6185308-B1) – key recovery

b. Stanton et al (US-6246771-B1) – session key recovery

c. Young et al (US-6282295-B1) – ARC key recovery

d. Kaliski, JR (US-2001/0055388-A1) – overall elements

e. Whiting et al (US-5778395-A) - backup

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Oscar Louie whose telephone number is 571-270-1684. The examiner can normally be reached Monday through Thursday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, James Myhre, can be reached at 571-270-1065. The fax phone number for Formal or Official faxes to Technology Center 2100 is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

OAL
05/15/2007

James Myhre
Supervisory Patent Examiner